Oligo-Snoop: A Non-Invasive Side Channel Attack Against DNA Synthesis Machines

Sina Faezi*, Sujit Rokka Chhetri*, Arnav Vaibhav Malawade*, John Charles Chaput*, William Grover†, Philip Brisk†, and Mohammad Abdullah Al Faruque*
*University of California, Irvine, Email: {sfaezi, schhetri, malawada, john.chaput, alfaruqu}@uci.edu
†University of California, Riverside, Email: wgrover@engr.ucr.edu, philip@cs.ucr.edu

***Abstract*—Synthetic biology is developing into a promising science and engineering field. One of the enabling technologies for this field is the DNA synthesizer. It allows researchers to custom-build sequences of oligonucleotides (short DNA strands) using the nucleobases: Adenine (A), Guanine (G), Cytosine (C), and Thymine (T). Incorporating these sequences into organisms can result in improved disease resistance and lifespan for plants, animals, and humans. Hence, many laboratories spend large amounts of capital researching and developing unique sequences of oligonucleotides. However, these DNA synthesizers are fully automated systems with cyber-domain processes and physical domain components. Hence, they may be prone to security breaches like any other computing system. In our work, we present a novel acoustic side-channel attack methodology which can be used on DNA synthesizers to breach their confidentiality and steal valuable oligonucleotide sequences. Our proposed attack methodology achieves an average accuracy of 88.07% in predicting each base and is able to reconstruct short sequences with 100% accuracy by making less than 21 guesses out of 4^{15} possibilities. We evaluate our attack against the effects of the microphone's distance from the DNA synthesizer and show that our attack methodology can achieve over 80% accuracy when the microphone is placed as far as 0.7 meters from the DNA synthesizer despite the presence of common room noise. In addition, we reconstruct DNA sequences to show how effectively an attacker with biomedical-domain knowledge would be able to derive the intended functionality of the sequence using the proposed attack methodology. To the best of our knowledge, this is the first methodology that highlights the possibility of such an attack on systems used to synthesize DNA molecules.**

I. Introduction

The ability to rapidly sequence and synthesize DNA has profound implications for society. Large libraries of different DNA sequences play an essential role in genomics research, especially for genetic analysis. Synthetic DNA is poised for widespread consumption if its costs can be lowered dramatically. Based on current trends, the global market for synthetic biology is projected to reach $38.7 billion by 2020 [58]. Beyond biological applications, researchers are beginning to construct DNA-based archival storage systems, which can store up to 215 petabytes of data per gram, with centuries to millennia of endurance if properly stored in a cool and dry environment [52].

Unfortunately, technological advancement often creates new security concerns as technologies mature. To date, the foremost security threat in this field involves the physical safety of synthesized DNA. Present efforts to reduce or eliminate misuse of synthetic DNA include biosecurity regulations, training and licensing programs for authorized agents, and the embedding of screening chips into DNA synthesizers (modeled on parental control of television access) [47], [11], [54]. However, these threat models implicitly assume that the value is inherent in the DNA itself, as opposed to the information that is encoded in the DNA.

Somewhat more generally, the cyber-physical nature of biotechnology workflows creates new security risks, which the corresponding research community has mostly neglected [48]. One recent example is the now-demonstrated ability to encode information into a DNA sequence that can trigger a buffer overflow error in DNA sequencing software; this exploit can be used to inject malware into the computer running the sequencing algorithm [46]. A subsequent concern is the confidentiality of DNA sequences stored in human biobanks. If the genetic information of the earth's population is exposed, then an attacker may be able to create a contagious virus that is fatal to individuals or a small group, but is otherwise benign to the general population [45].

Confidentiality concerns also extend to synthetic DNA sequences. In synthetic biology, the objective is often to engineer an organism with desired traits or functions. Investors only reap the rewards of their investments *after* the engineered organism passes all regulatory requirements and the investor obtains intellectual property ownership in the form of a patent or copyright. However, while the organism is still under development, the research remains vulnerable to industrial espionage or academic intellectual property theft [59]. In this case, the actual secret to be protected may be an amino acid sequence within a protein (which is derived from DNA) as opposed to the DNA itself. Within this larger context, knowledge of the DNA can still help an attacker determine the amino acid sequence, and the attacker can further benefit if he or she has knowledge of the desired traits or functions of the organism under development.

A. Motivation and Overview

This paper presents *Oligo-Snoop*: a novel, acoustic, side-channel, analysis-based attack model that can breach the confidentiality of DNA synthesizers. The attack model leverages the physical implementation of the synthesizer to infer the DNA sequence being synthesized. By publishing this attack, we hope to encourage commercial DNA synthesizer manufacturers to

Network and Distributed Systems Security (NDSS) Symposium 2019
24-27 February 2019, San Diego, CA, USA
ISBN 1-891562-55-X
https://dx.doi.org/10.14722/ndss.2019.23544
www.ndss-symposium.org

strengthen their confidentiality, especially to protect against attack vectors that may be discovered in the future.

As a motivating example, Gibson *et al.* synthesized the genome of a living bacterium out of one million bases of synthetic DNA [29]; eavesdropping on that DNA synthesis run would provide the attacker with the blueprints of a complete organism. More often, instead of synthesizing an entire genome from scratch, researchers add synthetic DNA to an existing organism's genome, thereby imparting desired traits to that organism. For example, for many years the anti-malaria drug artemisinin was available only from a rare plant; however, in 2006 Ro *et al.* added DNA to yeast cells, inducing the modified yeast to produce artemisinin [51], which dramatically reduced the cost of producing a lifesaving drug. In a more recent (and rather controversial) example, Galanie *et al.* added DNA to yeast cells to force them to create prescription opioid drugs [28]. Synthetic DNA plays a key role in each of these examples, and for these and similar efforts to remain secure, it is necessary to develop further protection against eavesdropping.

From a different perspective, the ability to eavesdrop on a DNA synthesizer could be useful in the fight against bioterrorism. Although DNA synthesis has several beneficial applications, there are many ways that it can be used maliciously. Since pathogens are composed of DNA, synthesis methods can be used for artificial pathogen creation. For years, researchers and government agencies have warned that an aspiring terrorist could use synthetic DNA and the techniques of synthetic biology to create deadly pathogens [11] . For example, the deadly Ebola virus has a genome of only about 18,960 bases [12] and could be built from scratch using synthetic DNA, as could genes from the eradicated disease smallpox, which was responsible for 300-500 million deaths in the 20th century alone [63] . The risk of synthetic, pathogenic DNA is significant enough that in 2010 the US Department of Health and Human Services issued a statement to commercial DNA synthesis companies, warning them to be on the lookout for customers ordering "sequences of concern," or snippets of DNA from the genomes of anthrax, Ebola, smallpox, and several other deadly pathogens [1]. With second-hand DNA synthesizers available on the online auction site eBay for less than $1000, it is feasible that an aspiring bioterrorist could try to use synthetic DNA to create their own tools of biological warfare. The ability to eavesdrop on a suspected terrorist's DNA synthesizer could potentially allow an intelligence or law enforcement officer to ascertain whether or not the suspect is trying to manufacture a deadly biological weapon.

B. Research Challenges

Technical challenges associated with DNA synthesizer confidentiality are as follows:

- Understanding the DNA synthesis process and its physical implementation.
- Identifying vulnerable components of a DNA synthesizer which can be leveraged under a practical threat model.
- Analyzing attack methodologies which an attacker may utilize.
- Understanding the ways in which an attacker may post-process side channel data to accurately reconstruct the DNA sequences that were synthesized.

C. Technical Contributions

This paper makes the following technical contributions, which directly address the challenges listed above:

- We provide a feasibility analysis (**Section IV**) to identify potential sources of side-channel information leakage in the system which have never been considered before.
- We present an attack model and propose a practical design approach (**Section III** and **V**) that an attacker may use to breach the confidentiality of the DNA synthesizer and reconstruct the DNA sequences using information leaked by the acoustic side-channel.
- We propose an algorithm (**Section VI**) that allows an attacker to obtain the other most likely reconstructions of the synthesized DNA sequence if the originally reconstructed DNA sequence is faulty.
- Due to the uniqueness of the proposed attack, in **Section VII**, we propose new methods to evaluate our work in terms of performance. For instance, in **Section VII-E** we show how to model the distance between the DNA synthesizer and microphone without exhaustive experimentation.
- We propose using a free tool designed for a different purpose to map imperfect attack model predictions onto more meaningful DNA sequences.

D. Paper Organization

The paper is organized as follows: Section II presents the background necessary to understand the system and process and summarizes related work on confidentiality in cyber-physical systems; Section III presents the threat model used to breach DNA synthesizer confidentiality; Section IV analyzes potential sources of acoustic emissions from the synthesizer; Section V presents the proposed attack methodology; Section VI explains how an attacker can reconstruct synthesized DNA sequences from acoustic measurements with a small number of guesses; Section VII reports experimental results; Section IX discusses potential countermeasures to the attack; and Section X concludes the paper.

II. Background & Related Work

A. Oligonucleotide Synthesis

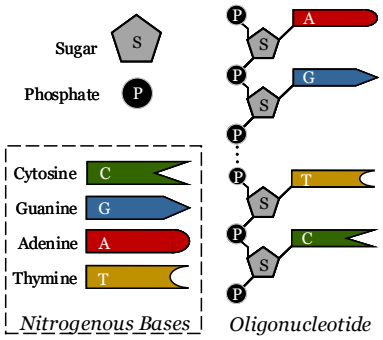


Fig. 1: Nucleotide bases and oligonucleotide sequence.

Oligonucleotides are the building blocks of DNA and RNA molecules. As shown in Figure 1, an oligonucleotide is a sequence of nucleotides. Each nucleotide comprises one of four nitrogen-containing nucleobases (Adenine (A), Cytosine (C), Guanine (G), and Thymine (T)) attached to a sugar (deoxyribose) and a phosphate group. The oligonucleotide is

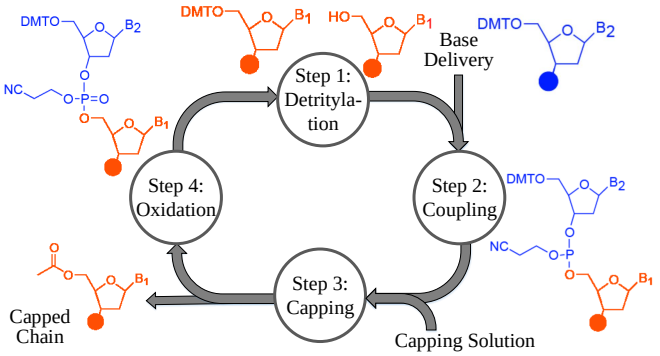


Fig. 2: Oligonucleotide synthesis cycle.

formed by constructing an alternating sugar-phosphate backbone, which joins the nucleotides to one another in a chain of covalent bonds. DNA, which is double-stranded, is formed by joining two complementary oligonucleotides according to base pairing rules (A with T and C with G) where complementary base pairs are joined by hydrogen bonds.

The term "DNA synthesis" is somewhat of a misnomer: so-called DNA synthesizers typically produce oligonucleotides, not double-stranded DNA. If DNA is desired, the user must synthesize two complementary oligonucleotides (typically multiple copies of each) and induce bonding via chemical or enzymatic means. Oligonucleotide synthesis produces short chains of nucleic acids with a defined sequence of bases. The most common form of oligonucleotide synthesis uses the phosphoramidite method [44], which produces multiple chains simultaneously by anchoring bases to a solid support and building upwards. DNA or RNA molecules have groups of 5 carbon atoms in the deoxyribose backbone. These carbon atoms are numbered 1' to 5'. While building the chains, a protective dimethoxytrityl (DMT) group is attached to the open 5' end of each chain. This prevents them from reacting or bonding with undesired materials before a new base is attached.

Figure 2 illustrates the process of adding a new base to an oligonucleotide:

- **Detritylation**: The protective DMT groups of the current chains are stripped away so the 5'-terminal can bond to the next base.
- **Delivery**: The next nucleoside phosphoramidite base to be attached is delivered to the solution.
- **Coupling**: A coupling agent, which contains a catalyst that causes the nucleoside to bond with the existing oligonucleotide, is delivered to the solution.
- **Capping**: A small percentage of the chains do not react in the coupling stage and thus do not receive a new base. The base support holding the oligonucleotide is treated with a capping solution that suppresses the addition of further nucleosides.
- **Oxidation**: The attachment point between the current oligonucleotides and the newly added base takes the form of a tricoordinated phosphate triester linkage. This structure is not natural and has limited stability. To improve the stability of this attachment point, the oligonucleotides are treated with iodine and water in the presence of a weak base to oxidize the phosphate triester, transforming it into a tetracoordinated phosphate triester. This form of linkage is natural, stable, and protected.

The oligonucleotides are now ready to receive their next base. The process repeats for every new base addition. Once all the bases have been attached, the oligonucleotides are cleaved from their solid support structures and collected for use.

B. DNA Synthesizer

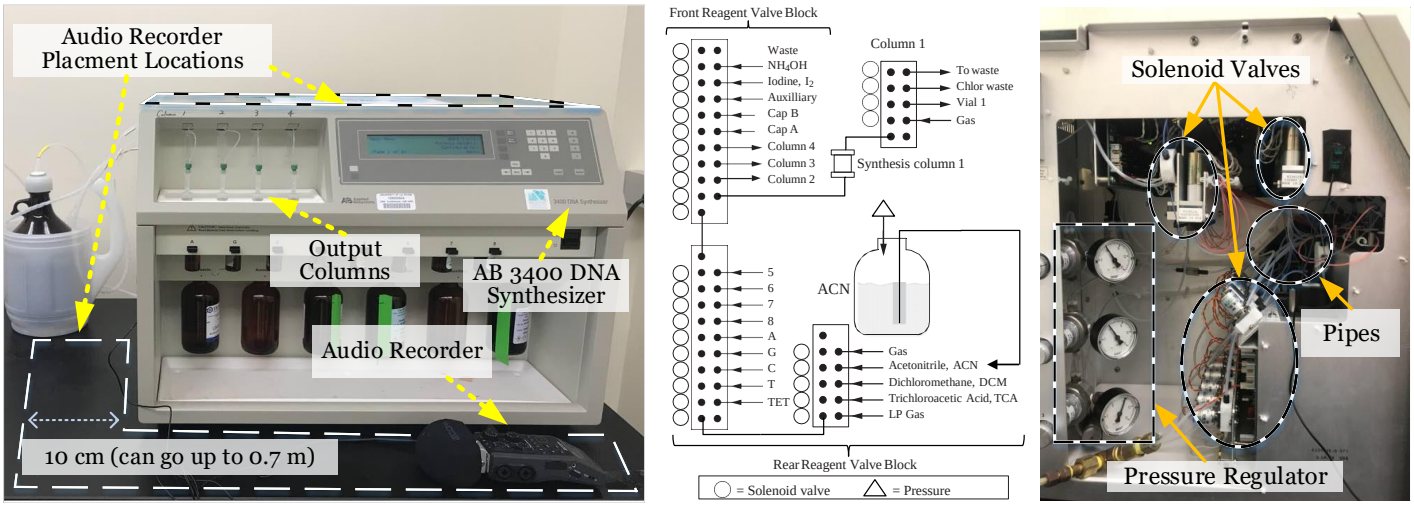
DNA synthesizers use a pressure-driven system, shown in Figure 3, to deliver chemicals to the output columns where synthesis takes place. A pressurized inert gas pushes chemicals through common pathways and delivers them to the synthesis columns. Blocks containing solenoid valves open and close certain pathways to route chemicals to the synthesis columns. During each iteration of the synthesis procedure, the common pathways are flushed to remove any leftover residue from the previous iteration.

Our experiments use an Applied Biosystems (AB) 3400 DNA Synthesizer [3]. This specific machine performs 48 valve operations during each synthesis cycle (Figure 2). Each step of the synthesis cycle requires multiple valve actuations to clean and prime the delivery lines before issuing step-specific valve operations that deliver the requisite chemicals to the columns.

Some large-scale DNA synthesis machines deliver multiple bases per delivery operation or deliver the same base to different columns at the same time. These machines can complete batch synthesis operations with higher throughput than single-operation synthesis machines such as the AB 3400. However, they still follow the same sequence of steps as shown in Figure 2 and produce similar results, albeit in larger quantities.

C. Side-channel and Information leakage

Side-channel analysis has been studied extensively in various systems to determine their vulnerability. Analog emissions (acoustics, power usage, electromagnetic emissions, vibrations, etc.) are one of the primary side-channels known to leak information. Acoustic emissions have been used to infer fill patterns in additive manufacturing systems [6], [14], [15], [24] and carry out physical attacks on magnetic hard disks [9]. Authors in [61] provided extensive results on the usability of light, seismic and acoustic side-channels by providing their channel characteristics such as rate and path loss. A power side-channel was used in [17] to detect malware in medical embedded systems. Authors in [67] utilized a memory side-channel to detect the activity of unwanted co-resident's virtual machine and the authors of [68] presented cache-based side-channel attacks which can be mounted on existing commercial clouds to steal cross-tenant information. Electroencephalography (EEG) signals obtained from a brain-computer interface were used in [42] to infer private user information. Authors in [13] demonstrated how network traffic based side-channels can be quantified for securing web applications. Timer interrupts and cache based side-channels were used in [32] to achieve a higher success rate than page-fault based side-channel attacks on untrusted operating systems. Authors in [4] demonstrated how accelerometer based side-channels can be used to infer user login details on smartphones. Each of these works demonstrates how various side-channels can be utilized to either infer information or provide better defense mechanisms in various systems.



(a) (b) (c)

Fig. 3: (a) Experimental setup. (b) Simplified schematics of a DNA synthesizer. Refer to [3] for a detailed explanation of this schematic. (c) The internal structure of the DNA synthesizer.

D. Acoustics

Vibration of a system in contact with air molecules generates a mechanical wave called an acoustic signal. Commonly, the intensity of this acoustic signal is expressed in terms of sound pressure level (SPL) using

$$L_p = 20log(\frac{p}{p_0}), \tag{1}$$

where p is root-mean-square (RMS) of acoustic pressure and p_0 is the minimum hearable acoustic level by human ears [25]. For a quiet office $L_p = 50$ dB; normal conversations $L_p = 60$ dB; vacuum cleaner $L_p = 70$ dB; and hair dryer $L_p = 80$ dB. Human ears barely detect a 3 dB difference in SPL while a 5 dB change can be easily noticed under most conditions.

III. Attack Model

Figure 4 depicts an attack model that can breach the confidentiality of a DNA synthesizer via information leaked in the acoustic side-channel. The components of the attack model are described as follows:

Adversary Intent: Industrial espionage for stealing intellectual property (IP) and monitoring for bioterrorism activities are covered by this attack model but speak to the intent rather than the identity of the attacker.

Outcome of an Attack: The attacker recovers the sequence of states of the target DNA synthesizer, $\hat{S}_{target}$, which translates to the order and types of the bases that are synthesized by the machine.

Target System: DNA synthesizers can connect to computers, external drives, and Ethernet cables. However, operators generally keep the machine disconnected from the Internet and local networks or use secured protocols to eliminate the possibility of cyber-attacks. In addition, we assume that tampering with the machine or accessing the output DNA sequence is not possible so the attack must be non-invasive. This is because fluids in the machine are sealed and are driven by pressurized argon. Any exposure to air would result in significant quality degradation, raising an alert.

Existing Vulnerabilities of the System: The DNA synthesizer is vulnerable to physical emissions that can leak system data during operation. Minimizing system observability can hide the system states from the attacker; however, DNA synthesizers are optimized for efficiency (throughput) and accuracy, not security.

Attack Medium: The attacker acquires information about the state of the DNA synthesizer via acoustic side-channel (A).

Attacker Capabilities: We assume that the attacker has the capability to place at least one microphone within close physical proximity to the DNA synthesizer. Such an attacker could be a disgruntled employee or a visitor with low-level access to the machine (meaning physical proximity, and no access to cyber components). In this scenario, the attacker can surreptitiously and non-intrusively place an audio recording device (such as a phone) near (or on) the DNA synthesizer. Placing the recorder requires one-time access if we assume that it has wireless transmission capabilities; otherwise, a second physical visit is required to recover the recorder and the data it has collected. Since the authorized users of the target DNA synthesizer are unaware of the attack model introduced in this section, most probably, they would neglect the security implications of any recording device around the machine. Furthermore, if an attacker is able to breach the other systems in the same laboratory (i.e. remote monitoring systems [56], employee phone/laptop, etc.), he will be able to record the information leaked in the acoustic side-channel of the DNA synthesizer through existing microphone(s) of those systems.

Attacker Resources: We assume that the attacker not only has domain knowledge about the synthesis process, but also has access to the user manual of the machine explaining the machine specific procedures for DNA synthesis. Furthermore, we assume that the attacker has the opportunity to carry out as many experiments as needed to *profile* the machine and build an accurate model ($\hat{S} = \hat{f}(A)$, see Figure 4) that can infer the order of synthesized bases based on recorded acoustic signals collected from the target machine. If the attacker is a disgruntled employee who has unlimited access to the target

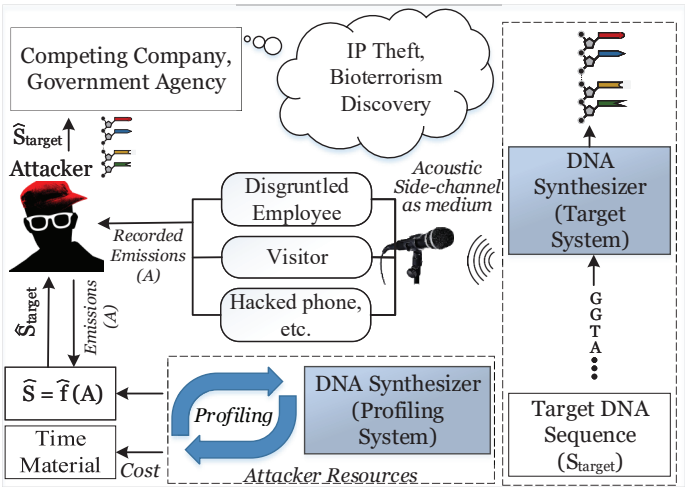


Fig. 4: Attack model.

system, the profiling DNA synthesizer can be same as the target machine. However, if access is not provided, the attacker could use a replica for profiling purposes (the same model with a structure that is identical to the target machine). For the rest of this paper, we consider the former scenario where the target and profiling DNA synthesizers are the same machine.

Cost: For an attacker to profile the target machine, the cost of an attack is just the value of the chemical materials used and the time spent during the profiling process (estimating ($\hat{f}$)).

IV. Feasibility Analysis

While oligonucleotide synthesis is in progress, the physical activity of different components of the system such as solenoid valves, cooling system fans, pressure regulators, and fluids flowing in pipes causes vibration which results in structural acoustic noise emission from the system. We hypothesize that various solenoid valves opening/closing and the flow of fluid through various pipes emit information about the various states of the oligonucleotide synthesis cycle, and that we may be able to identify which nitrogenous bases (A, G, C or T) are deposited during the delivery state of the oligonucleotide synthesis cycle. An attacker may thus eavesdrop on the acoustic emissions, that behave as side-channels, to infer the cycles and the type of the base being delivered.

A. Structural Acoustics caused by the pipes

DNA synthesizers use plastic pipes (lines) to deliver the nucleotide bases and other chemical materials from the source reservoirs to the output columns and other containers attached to the system. The internal turbulence of the fluid flow running in the pipe causes the walls of the pipe to vibrate, resulting in acoustic noise (i.e. vibro-acoustic) radiation from the pipes. Over the last few decades, a substantial body of research has been dedicated to modeling fluid-structure interactions to simulate and predict vibro-acoustic signal emissions of pipe structures [40], [64]. Work in this area has shown that the magnitude and frequency of a generated vibro-acoustic signal can be determined based on the spatial structure of the pipes, pipe wall thickness, internal pipe pressure, internal fluid speed, the mass density of the fluid and the pipe, and several other features. Authors in [60] have demonstrated how minute changes in the curvature of the elastic pipes can result in different vibro-acoustic footprints. As shown in Figure 3c, the delivery lines in the DNA synthesizer have different spatial shapes and curvatures and also deliver fluids with different mass densities. Based on the work in [60], we suspect that these changes will result in close but unique wavenumbers.

B. Structural Acoustics caused by the solenoids

DNA synthesizers employ electric solenoids to open and close the valves that control the flow of chemicals to each column; each valve opening or closing operation emits an audible click. Although each solenoid emits a near-identical sound, the DNA synthesizer is an enclosed structure and each valve is located at a different position within the machine. As stated in [65], the enclosed space creates measurable reverberations when a valve emits sound. The equation to calculate reverberation time is given in [65] as:

$$T = 0.049\frac{V}{Sa}, \tag{2}$$

where V is the volume of the enclosure, S is the surface area which reflects sound, and a is the average Sabine coefficient of the enclosure. As each valve occupies a distinct position within the DNA synthesizer, the surface area that causes reflections, which impacts the reverberation time, is unique for each valve. Consequently, the collected acoustic signals are likewise unique for each valve due to their unique channel distortions. Similar to the sound generated from the fluids moving through the machines piping, the distinctions between valve noises may be near-inaudible for human listeners. However, a properly trained algorithm should be able to identify key features that distinguish different valve operations.

V. Attack Model Design

Here we present the design of the attack model, which we introduced earlier in Section III. In order to accurately infer the physical and cyber-domain states of the system (S) from the acoustic side-channel (A), an optimal attack could first use principle-based equations to derive a function ($A = f(S)$) to explain the sound produced by the individual components of the system based on the DNA sequence. Then, using techniques like finite element analysis, the attacker may acquire an accurate acoustic emission profile of the DNA synthesizer. Afterward, the attacker may use the inverse function to estimate the sequence ($\hat{S} = f^{-1}(A)$). However, this approach would require an attacker to have complete design details of the individual components, their chemical composition, etc., to accurately simulate the acoustics from the system. To overcome this problem, we propose to use a data-driven approach, by treating the DNA synthesizer as a black-box, to estimate the function ($\hat{S} = \hat{f}(A)$). This approach requires less domain knowledge and achieves faster attack model implementation time for an attacker.

As shown in Figure 5, to estimate the function that describes the relationship between the acoustic signal and the oligonucleotide sequences, our proposed attack model consists of two main phases: the training phase and the attack phase. This function may be abstracted as $S = \hat{f}(A, \theta)$, where θ is the parameter that needs to be trained,

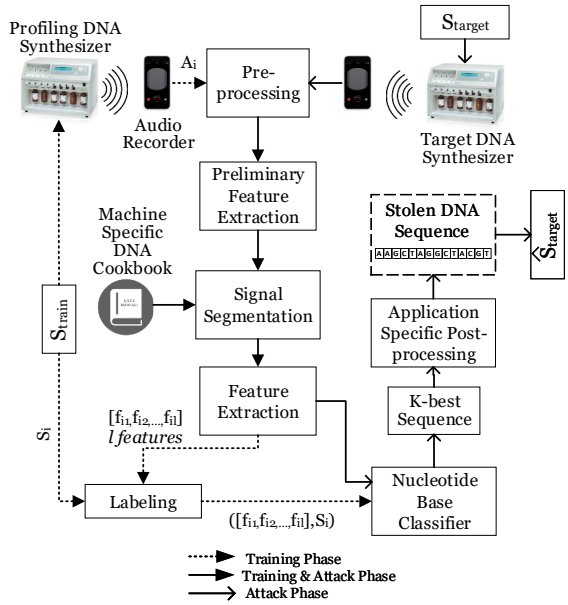


Fig. 5: Acoustic side-channel attack methodology.

$S = [S_1, S_2, \ldots, S_n], S_i \in \{A, G, C, T\}$ is the sequence of oligonucleotides with length n, and A represents the acoustic signal gathered from the side-channel. In the training phase, an attacker randomly selects an arbitrary number of training oligonucleotide sequences (S_{train}). These sequences are then passed to a profiling DNA synthesizer. Then for each of the $S_i \in \{A, G, C, T\}$, the attacker collects the corresponding acoustic emission ($A_{i_1}, A_{i_2}, \ldots, A_{i_k}$), where k is the length of the acoustic emission. The attacker must initially find an optimal location to place the acoustic sensors, which, in our work, are placed next to the DNA synthesizer in close proximity to the solenoids and the pressure valves. We then perform preprocessing and feature extraction on these acoustic emissions and label them with their corresponding nucleotide bases $\{A, G, C, T\}$. Using a supervised learning approach [53], a classifier function is estimated to predict the particular nucleotide base given the acoustic emission $\hat{S}_i = \hat{f}(A_i, \theta)$. In the attack phase, the attacker surreptitiously places sensors on the target DNA synthesizer and collects the acoustic emissions, and infers the target oligonucleotide sequence (S_{target}). The details of the each of the steps of the attack model design are as follows:

Preprocessing. In this stage, the attacker uses a set of bandpass filters combined with heuristic methods to reduce the effect of background environmental noise, which is added to the acoustic signal generated by the DNA synthesizer. For instance, the attacker may use

$$A_{normalized} = diag(\frac{1}{\sqrt{diag(Rnn) + \epsilon}} \times A) \qquad (3)$$

similar to what it has been used in [36] to model the background noise and normalize the signal in relation to it. In this model, Rnn is the background noise covariance matrix based on a portion of a recording when the machine is idle; A is the recorded signal, and $\epsilon = 1 \times e^{-10}$ is used to avoid division by zero. In our experience, the background environmental noise is usually more prominent in lower frequency ranges. Hence, if the attacker determines that the DNA synthesizer does not leak information in lower frequencies, he/she can simply use a high-pass filter to eliminate low-frequency components from the signal.

Preliminary feature extraction. Once background noise is removed from the recorded signal, the attacker needs to extract the portions of the signal that correspond to base deliveries $\{A, G, C, T\}$. As shown in Figure 2, the oligonucleotide synthesis process goes through various stages, base delivery being one of them. An attacker needs to know the time taken by various stages in order to accurately segment the acoustics for just the nucleotide base delivery stage. As shown in Figure 6, opening and closing the solenoid valves introduces peaks in the acoustic signal. These peaks may help an attacker track various stages.

However, difficulties may arise when multiple valves open and close at the same time, which would result in peaks with variable intensity. To mitigate this issue, an attacker could use an approach proposed in [20] to detect the peaks by using their shape with the help of wavelet transforms. Furthermore, since the properties of the solenoid valves are assumed to be known to the attacker, he/she can specify the minimum distance between peaks to increase the accuracy of the peak detection algorithm.

Signal segmentation. Since an attacker has access to the user manual for the DNA synthesizer, he/she knows duration of each stage. Then, using the peak detection algorithm and the timing data from the user manual, the attacker can segment the nucleotide delivery stage. Since all the other stages of the synthesis remain the same for adding each new base, an attacker only needs the base delivery stage to reconstruct the sequence. Although this step can be done manually for shorter sequences, an attacker who wishes to reconstruct long DNA sequences may consider more sophisticated techniques, such as Hidden Markov Models (HMMs) [21] or Long Short-Term Memory (LSTM) neural networks [35], which have historically been used in voice recognition, to track the different states of the machine. Since the relative distance between the peaks is a known constant (as is described in the user manual), both of these models will achieve high accuracy; however, our experience has shown that neither of these models is perfect, and that a successful attacker will need to manually segment the data to obtain 100% accuracy. 100% accuracy is needed, since any error in this stage will jeopardize the attack model's subsequent steps.

Feature extraction. As shown in Figure 6, the base delivery segment of the signal consists of three sections: the first peak, which is the result of opening the valve that controls the flow of a certain base to the output column; a longer section in which the pipes deliver the base; and a final peak, which is the result of closing the valve whose opening generated the first peak. Since the duration of each solenoid valve operation is known, the attacker can divide the delivery segment into three sections and engineer a specific set of features for each. Extractable features range from simple calculations such as the standard deviation of the signal to complex calculations such as coefficients of Fourier and wavelet transforms. Since the length of the signal is short for a delivery segment (less than 5.5 seconds in our experiments), we can assume that the attacker has enough computational power to calculate any of these features for all three sections of the delivery segment. However,

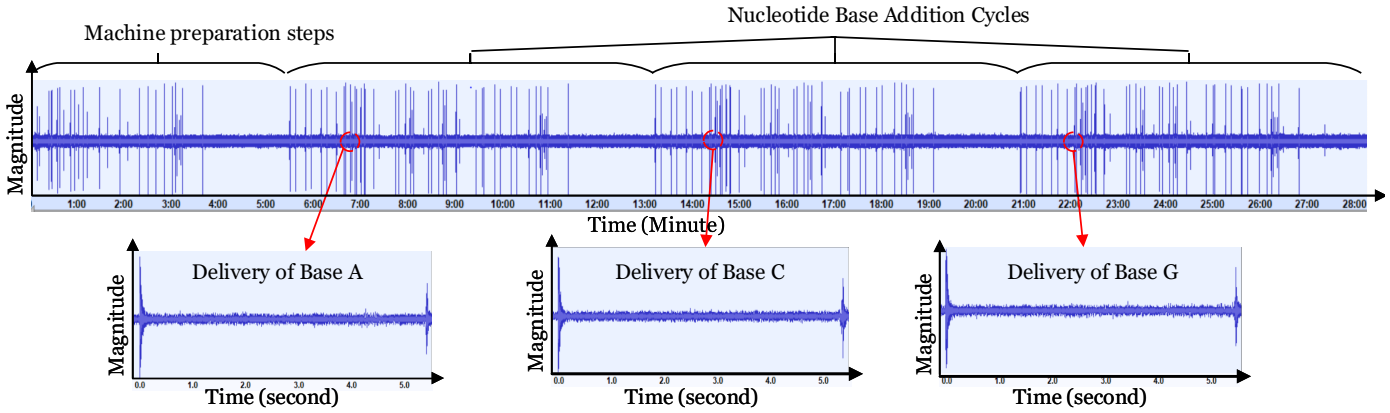


Fig. 6: Sample acoustic signal emission from DNA synthesizer.

extracting all possible features will significantly affect the convergence rates of the classifiers that will use them. Hence, in the training phase, the attacker creates models using a subset of all available features, either by feature projection (e.g. PCA [62], LDA[41]) or feature selection (e.g. [16], [5]). Based on our experiments, the latter approach works much better for acoustic side-channel attacks on the DNA synthesizer because even small environmental background noises mask most of the useful features when PCA or LDA projects them onto lower dimensions. The outcome of this stage is the conversion of an acoustic signal $(A_{i_1}, A_{i_2}, \ldots, A_{i_k})$ into a set of features $(f_{i_1}, f_{i_2}, \ldots, f_{i_l})$ with $l \ll k$.

Nucleotide base classifier. In this stage, the attacker selects and trains the best classification algorithm to estimate the function $(S_i = \hat{f}(f_{i_1}, f_{i_2}, \ldots, f_{i_l}, \theta))$ that correlates a given set of features to one of the four nucleotide bases. To find the best algorithm, he/she trains multiple classifiers such as neural networks [34] and random forests [10] and calculates the accuracy of each classifier. Each of these functions will have a certain method of training $\theta = \{\theta_1, \theta_2, \ldots, \theta_m\}$, where m depends on the type and architecture of the classification algorithm used. The accuracy of a classifier is defined as the percentage of correct predictions divided by the total number of predictions made over the test data-set. To ensure that enough training samples have been provided to the models, the attacker monitors the corresponding accuracy of predictions in terms of the number of training points. If the accuracy stops improving when new samples are added to the training dataset, the attacker can assume that the classifiers have converged. Once the attacker identifies the most accurate classification algorithms, then he/she can use an ensemble of algorithms to improve the accuracy even further. To do this, the attacker computes the normalized probability distribution for each base (A, G, C, T) for each classification algorithm and selects the most probable base as a result. The caveat of having a classifier only predict a single nucleotide base is that an attacker still needs to reconstruct the whole chain, which can introduce additional complexities. In order to tackle this issue in our attack methodology, we propose using an algorithm that produces the k-best sequences of oligonucleotides based on the output of the nucleotide classifier. The details of the k-best algorithm are presented in Section VI. In the training phase, an attacker constructs a nucleotide classifier; in the attack phase, the attacker can use the k-best sequencing algorithm, along with domain-specific post-processing, to reconstruct the sequence. Finding the first best sequence is trivial since the attacker only needs to choose the type with the highest probability for each delivery to achieve the highest confidence in the whole sequence prediction. However, finding the next best sequences are not straightforward, and he/she can use the *K-best DNA sequences* algorithm.

Post-processing. The classification algorithms described in the previous stage provide results based on the features present in the given segment of the signal; they ignore any relation of the current base delivery to past or future base deliveries. In practice, the order of the bases in an oligonucleotide sequence follows certain rules based on the synthesis technology, machine capabilities, and the specific domain for which it is being synthesized. For instance, authors of [46] identify three major limitations for DNA synthesis. The first limitation involve homopolymers, which are repeated sequences of the same base; this eliminates the chance of synthesizing an oligonucleotide sequence more than $\sim$10 consecutive instances of the same base as a substring. The second limitation is that a reasonable ratio of G and C bases should always appear in the sequence. The last limitation involves secondary structures, in which an oligonucleotide sequence contains multiple complementary subsequences that may bind to one another, creating a physical loop. In addition, domain-specific knowledge can improve the accuracy of predictions. For example, there are well-known sequences in synthetic biology that are responsible for certain functions. An attacker with this knowledge will be able to correct errors in the algorithm if he/she notices similarities in the extracted sequences to those mentioned. The same can be found in most other applications of synthesized DNA. For example, in the case of storing data in DNA molecule, if the data is coupled with error detection/correction bits, then identifying the errors will be possible for the attacker as well.

The classification and domain-specific post-processing schemes report a reconstructed oligonucleotide sequence, which the attacker has effectively stolen. Further laboratory experiments can then confirm the correctness of the reconstructed sequence. If the attacker concludes that the reconstructed sequence is incorrect, he/she will want to consider other probable sequences, until he/she discovers a reconstructed sequence that he/she believes to be correct.

Algorithm 1 DAG generation algorithm.

Input: Probability distributions of base type prediction (P)
Output: Directed acyclic graph (G)
//node (<label>,<index>)
//edge (<from_index>,<to_index>,<weight>)
//Order of probability distribution rows in P : AGCT
procedure GENERATEDAG
    $n\_deliveries \leftarrow$ length of P
    $n\_nodes \leftarrow 4 \times n\_deliveries$

    //Creating the nodes
    $G \leftarrow$ node(start,-1), node(end,n_nodes)
    for $i = 0; i < n\_deliveries; i = i + 4$ **do**
        $G \leftarrow$ node(A,i),node(G,i+1),node(C,i+2),node(T,i+3)

    //Adding the first and last layer edges
    for $i = 0; i < 4; i++$ **do**
        $G \leftarrow \text{edge}(-1, i, P(0, i))$ //from source
        $G \leftarrow \text{edge}((n\_nodes-1)-i, n\_nodes, 1)$ //to end

    //Adding internal layers edges
    for $t = 0; t < n\_nodes - 4; t = t + 4$ **do**
        $i\_offset \leftarrow t$
        $j\_offset \leftarrow t + 4$
        for $i = 0; i < 4; i++$ **do**
            $i\_idx \leftarrow i\_offset + i$
            for $j = 0; j < 4; j++$ **do**
                $j\_idx \leftarrow j\_offset + j$
                $delivery\_id \leftarrow \frac{t}{4} + 1$ //next delivery
                $G \leftarrow edge(i\_idx, j\_idx, P(delivery\_id, j))$
    return G

VI. K-BEST DNA SEQUENCES

A nucleotide base classifier predicts $q = 4$ possible output classes (A,G,C,T) for each base. The classifier first estimates the conditional probability distribution of possible outputs ($Y = \{c_1, ..., c_q\}$) for the set of given input features f. Then, the classifier reports the class $S = c_k \in Y, k \in (1, 2, \ldots, q)$ with the highest probability as the result:

$$f \text{ is assigned to class } c_k \iff p(c_k|f) \geq p(c_r|f) \forall k \neq r. \quad (4)$$

where $r \in (1, 2, \ldots, q)$. The confidence of a classification algorithm for a certain prediction is defined to be equal to the probability of the predicted class. Since the prediction for each base delivery is independent from the others, the classifier computes the *confidence* value for the sequence as

$$Confidence = \prod_{i=1}^{n} p(c_{k_i}|f_i), \quad (5)$$

where $S_i = c_{k_i}$ is the predicted class for the i^{th} nucleotide base, based on the input features f_i. The *confidence* defined in Equation 5 represents the chance of predicting the target sequence with 100% accuracy. An attacker would want to maximize this value. Choosing a class such that $p(c_{k_i}|x_i) \geq p(c_{r_i}|x_i) \forall k_i \neq r_i$, will maximize the *confidence* value, thereby increasing the probability that the predicted sequence exactly matches the original sequence. However, as explained in the previous section, there exist scenarios where the attacker would prefer more candidate sequences in addition to the best-predicted one. In response, we propose an algorithm to predict the K-most probable orders of bases in a sequence by keeping the value of *confidence* as close as its possible to its maximum value. Our algorithm is inspired by the Viterbi algorithm [27] which is commonly used to find the most probable sequence of hidden states in an HMM for a given sequence of observations.

Algorithm 1 accepts as input a two-dimensional array P which contains the conditional probability distribution of the four base types with n delivery stages. The value of array entry $P(i, j)$ is equal to $p(c_j|f_i)$ where $c_j \in \{\text{A, G, C, T}\}$ and f_i is the given input (set of features $[f_{i_1}, f_{i_2}, \ldots, f_{i_l}]$) to the classifier for the i^{th} nucleotide base prediction. Algorithm 1 converts P into a Directed Acyclic Graph (DAG). The first step is to instantiate two dummy nodes to represent the beginning and end of the sequence. Second, four nodes with {A, G, C, T} labels are added as a layer between the start and end nodes to represent the four possible outputs of each classification step. The start node is connected to the four nodes labeled in the first layer by directed edges with weights set to $p(c_j|f_i)$, where c_j corresponds to the label of the destination node. The process repeats iteratively to add subsequent layers: the nodes in layer i are connected to the nodes in layer $i + 1$ by instantiating a directed edge with a weight equal to $p(c_j|f_{i+1})$. Directed edges with weight 1 are added from the four nodes in the final layer to the end node. The DAG enables simple and intuitive way to calculate the confidence of the reconstructed sequence.

Assumption 1: Algorithm 1 generates a DAG.

Remark 1: Algorithm 1 generates a graph layer-by-layer and only adds edges between layers i and $i + 1$.

Assumption 2: A path from the start to the end node in the DAG represents a candidate reconstructed sequence. The *confidence* of the corresponding sequence is equal to the product of the weights of the edges along the path.

Remark 2: A path in the DAG emanating from the start node will pass through each layer exactly once. Choosing the node's label as be the delivery base type yields a nucleotide sequence whose length is equal to the number of layers in the graph. All input edges to a node with a given label will weight equal to the probability of that label being correct for its corresponding base delivery. Considering that the value of the last edge in the path, which terminates at the end node, is 1, the product of the weights on the path is equivalent to the definition of the *confidence* shown in Equation 5.

We first call Algorithm 1 to generate DAG G from input P. We then replace all the weight values with their corresponding log base values for mathematical simplicity in later steps. We also define the length of a path to be equal to the summation of edge weights on the path. In this case, the k longest paths in the graph will represent the k-most probable sequences. Notice that the summation of the log of a set of values is equal to calculating the log of the multiplication of those values.

There exist multiple algorithms that compute the K longest paths from a source node to sink node in a DAG [22]. The algorithm presented in [23] achieves an optimal asymptotic time complexity of $O(m + nlogn + k)$, where n is the number of nodes and m is the number of edges in the DAG. This algorithm has an $O(1)$ time complexity per pathfinding attempt,

after a preprocessing stage that uses Dijkstra's Algorithm to identify the shortest path. This is a substantially more efficient than a brute-force approach, which would enumerate all 4^n length-n base sequences and compute their *confidence* values in $O(n)$ time per sequence, yielding an overall time complexity of $O(n4^n)$.

TABLE I: Probability distribution of base types for three consecutive deliveries.

Base Name	Delivery #1	Delivery #2	Delivery #3
A	**0.9**	0.03	0.12
G	0.05	**0.8**	**0.4**
C	0.01	0.15	0.35
T	0.04	0.02	0.13

Table I presents a sample probability distribution of base types for a DNA synthesis procedure consisting of three delivery stages. Based on this table it is easy to infer that the most probable sequence is AGG with *confidence* of $0.9 * 0.8 * 0.4 = 0.288$. However, since the probability of delivering base C in the last stage is very close to the probability of delivering base G, if the sequence does not support our requirements, then we would intuitively consider the sequence AGC. If the attacker determines that both AGG and AGC are incorrect sequences, then he/she would turn to Algorithm 1 to generate the DAG shown in Figure 7. A top-11 analysis of the DAG generates the following sequences, in order: AGG, AGC, AGT, AGA, ACG, ACC, ACC, ACT, ACA, GGG, GGC.

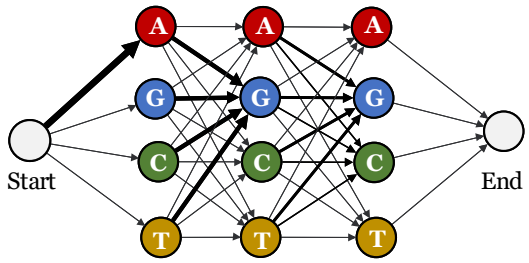


Fig. 7: A DAG corresponding to the probability distribution provided in Table I (Thicker arrows represents higher probabilities for the destination nodes.)

VII. Results and Case Study

This section presents the experimental results obtained by implementing the proposed attack methodology on an Applied Biosystems Inc. (ABI) 3400, one of the most widely used commercial DNA synthesizers. To validate this choice, we contacted a DNA synthesizer sales expert who stated: *"I strongly recommend you consider either the ABI 394 or the ABI 3400. They are by far the workhorses of the industry. 95% of our customers use the ABI 394."* - [Email]). This section also presents a test cases where we reconstructed the complete oligonucleotide sequences using the proposed K-best DNA sequences algorithm and post-processing steps.

A. Test Bed

As shown in Figure 3, the experimental setup consists of an AB 3400 DNA Synthesizer [3] and a Zoom H6 audio recorder to acquire the acoustic signal. We record the signals through three channels simultaneously at a sampling frequency of 48 kHz with a resolution of 24 bits per sample. For every DNA synthesis run, we randomly place a recorder with two condenser microphones near the DNA synthesizer (on top of the machine or on the setup desk, no further than 10 cm from the machine). We also use a contact microphone to record acoustic signals with almost no environmental noise. Our attack methodology is implemented in Python 3.6. We use the tsfresh [16] library package for feature extraction; scikit-learn [49] for profiling model generation; and networkx [31] for modeling the network discussed in Section VI. We also use MATLAB [43] to represent the Short-Time Fourier Transform (STFT) results.

B. Evaluations Assumptions

We used the Zoom H6 portable handy recorder tool kit, which has publicly available coil condenser microphone characteristics, to carry out our attack. The Zoom H6 is similar to an iPhone 4, which contains two similar internal microphones. As a ubiquitous consumer product, an iPhone 4 placed in a discreet location near a DNA synthesizer, would seem innocuous to the typical user of such a machine, and could easily collect days' worth of data without detection.

We assume that the DNA synthesizer is used exclusively for oligonucleotide synthesis. DNA synthesizers have various cycle scripts for producing different polymerases. If a user intends to synthesize DNA, the same script should be used, without modification, regardless of the target sequence; modification of the cycle script can cause erroneous synthesis behavior. (Over six months of studying this machine in an active biomedical laboratory, we observed that the settings were never changed. We verified that the cycle script that was run repeatedly by different users always matched the cycle script for oligonucleotide synthesis in the AB 3400 synthesizer manual. This assertion was subsequently confirmed by direct communication with the machine operators).

Although the AB 3400 can synthesize four columns in parallel, in practice, the deliveries to output columns do not occur simultaneously, as described in Chapter 6 of the user manual [3]; for each output column, the same solenoid valves and pipes are used. The only difference is setting the *Front Reagent Block* to a different output column before the next delivery cycle. Since the same script is used for every column, extracting the delivery stages for each output is straightforward. To avoid unnecessary complication, we focus on single output column DNA synthesis.

C. Training and Evaluation

The attack model described in Section V consists of functions ($\hat{f}(.,\theta)$) that need to be trained before they are used for a specific synthesizer. Since the objective of each model is known, we use supervised learning to estimate the functions. We initially synthesized seven different 60-base oligonucleotide sequences, each consisting of 15 A's C's G's and T's in varying orders. An attacker could increase the number of synthesis runs if the classification results do not converge, however, as shown in this section, the initial runs were sufficient. Each synthesis run took 7 hours, 29 minutes,

and 53 seconds. As an attacker, we label the acquired signals into different stages: 'initialization', 'repetitive cycle', and 'base delivery'. The labeling is possible because the user manual for the synthesizer machine lists the operations which take place during synthesis and the corresponding duration of each operation [3]. Based on the manual, it is easy to infer that the DNA synthesizer initialization stage takes approximately 787 seconds and only occurs at the beginning of the synthesis procedure. The 'repetitive cycle' stage takes approximately 463 seconds, and the 'base delivery' stage takes approximately 5 seconds inside the 'repetitive cycle' stage.

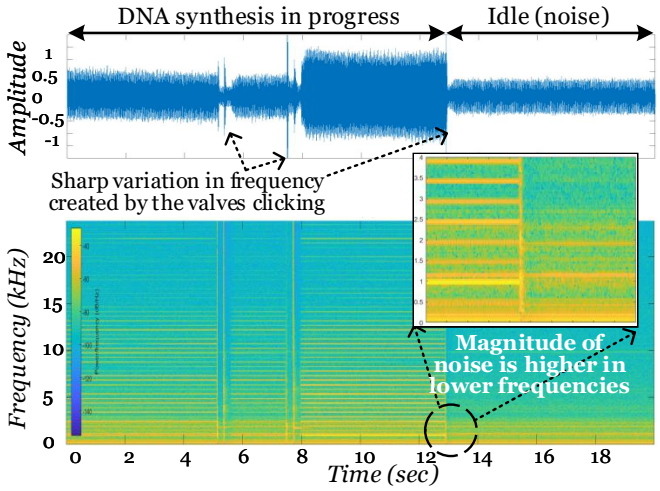


Fig. 8: Recorded acoustic signal from the DNA synthesizer during synthesis followed by an idle state (top). Short-Time Fourier Transform (STFT) of the corresponding signal (bottom).

To pre-process the signals, first the STFT spectrum is calculated. The analysis of the acquired signals reveals there is no difference in the magnitude of frequency components below 300 Hz between the portion of the signal which belongs to environment noise versus actual DNA synthesis (see Figure 8). Hence, we chose 300 Hz as a cutoff frequency and filter the signal frequency components below this limit using a high-pass filter. We apply Equation 3 to normalize the signal before further processing.

In Figures 6 and 8, each valve operation produces an audible click from the machine which is clearly visible as a peak in the recorded waveforms. When operational, the DNA synthesizer executes the cycle script, for which the sequence and duration of valve operations during the synthesis run are known. By correlating valve operation timings between the waveform and cycle script, sections of the waveform can be labeled with the corresponding valve operations. On the AB 3400 DNA synthesizer, the base delivery stage contains six valve operations with a unique sequence of timings relative to other stages in the synthesis process. With practice, the base delivery stage can be visually identified and extracted from the waveform. Additionally, the specific valve operation that delivers the base always occurs at the same time in the delivery stage. Therefore, this operation can be extracted and used for base identification in the classification step of the attack model. Since a human can manually extract the base delivery operation using these features, we implemented an algorithm to extract the base deliveries using the same process. First, we identify the peak locations in the signal using continuous wavelet transforms [20]. Then, based on the cycle script, the algorithm identifies the sequence of distances that correspond to the base delivery stage. For each stage, the algorithm references the cycle script again and extracts the segment that corresponds to the base delivery valve operation.

Once the base delivery segment is extracted from the signal (similar to what is shown in Figure 6), we train six classifiers as shown in Table II. Feeding the raw base delivery acoustic signal to these classifiers results in random classification ($\leq$ 25% accuracy), so a feature extraction step is required before classification. We select the best set of features to be used for classification in two steps. First, we extract all the features introduced with the tsfresh [16] library. These features consist of the time domain, frequency domain, and wavelet-based features. Next we calculate the significance of each feature and carry out multiple test procedures [7] to select the most relevant features with the lowest dependency score [16].

This procedure reduced the number of features from 57,018 to 75 (*selected features* in Table II). The selected features include the magnitude of the Fourier transform components of the input signal in certain frequencies as well as the autocorrelation of the signal with a lag of 2 and 3 samples. The selected set of features matches what was expected from the feasibility analysis of the attack described in Section IV. The structural differences between the pipes used for different base deliveries causes each base delivery to generate slightly different frequencies. However, for practicality, the tsfresh library with default settings does not generate all of the frequency components. To discover all of the possible features in the frequency domain while keeping computational resources fixed, we calculate the frequency components with an accuracy of 200 mHz, but only at frequencies above 300 Hz with local peaks in the frequency transform (see Figure 8). We reran the same feature selection algorithm and identified 310 features within those frequency bands (*improved selected features* in Table II).

To ensure that the amount of training data is sufficient for the models, we analyze the prediction accuracy based on the number of base delivery samples in the training dataset. We selected *AdaBoost* [33], *linear Support Vector Machine* (SVM) [18], *Naïve Bayes* [66], *Neural Network* [34], and *Random Forest* [10] classifiers to estimate the function $\hat{f}(.,\theta)$. We also implemented a weighted majority rule voting [50] based ensemble that uses *random forest* and *neural network* as the base classifiers; for simplicity we set the weights of both classifiers to be equal. As shown in Figure 9, around 200 samples is sufficient to train the models to achieve maximum accuracy when classifying nucleotide bases based on the selected features. We used 80% of the dataset for training and 20% for validation, coupled with 10-fold cross validation [38] to produce the results reported in Figure 9 and Table II. The reported accuracy numbers shown are averaged across the 10 folds. Figure 9 and Table II show that the majority rule voting-based ensemble of classifiers achieves faster convergence and higher classification accuracy than the other classifiers for the *improved selected features*. This can be explained by the fact that each individual classifier effectively searches a space $\mathcal{H}$ of hypotheses in

TABLE II: Accuracy of the classification models.

Classifier	Settings	Accuracy (%)			
		Raw Signal	All Features	Selected Features	Improved Selected Features
AdaBoost [33]	_	18.54	39.58	57.22	69.46
Support Vector Machine (SVM) [18]	kernel= 'linear', penalty_error(C) = 0.025	**26.8**	17.63	57.77	84.05
Naive Bayes [66]	_	12.12	27.63	58.61	75.31
Neural Network[34], [55]	architecture= 'fully connected feed forward', activation_func='relu', num_hidden_layers=100, num_training_iteration=1000, solver = 'adam'[37]	14.59	34.99	**64.44**	87.51
Random Forest [10]	num_estimators=100	23.4	46.66	60.69	78.46
Voting [50]	classifiers = {RandomForest, NeuralNetwork}	24.18	**51.25**	62.5	**88.07**

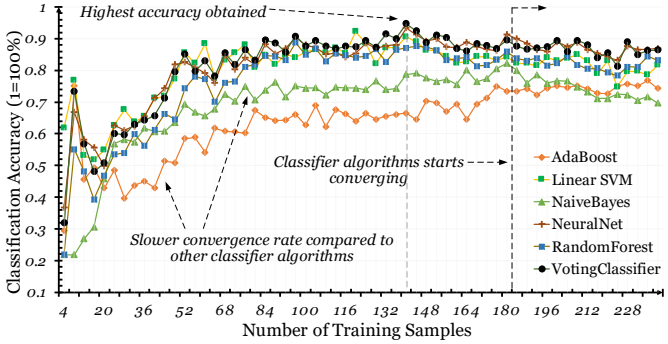


Fig. 9: Learning curve of various classifiers for nucleotide base classification.

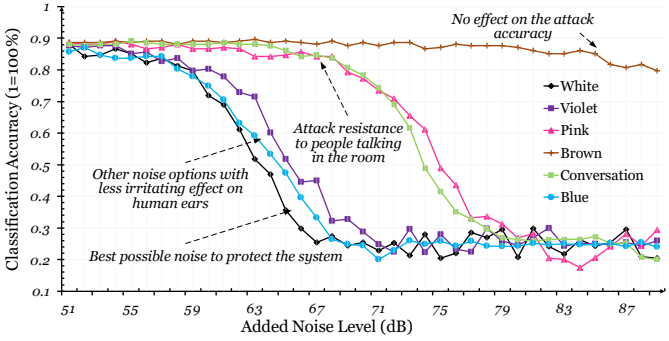


Fig. 10: The impact of added noise effects on the side-channel attacks against DNA synthesizers.

search of the hypothesis with the highest accuracy. Different classifiers identify different hypotheses with similar, if not the best overall, accuracy. If each classifier's hypothesis has uncorrelated errors with error rates exceeding 50%, then the voting-based ensemble method is likely to increase overall accuracy [19]; however, this improved performance is obtained at the cost of increased computing power and training time for the ensemble of classifiers used in voting.

The next step in the attack model, as discussed in Section V, is post-processing the reconstructed DNA sequence with domain-based knowledge after generating the K-best sequences. Since the accuracy of the classifiers is reported based on random delivery samples, we did not integrate this stage into our initial experiments. Instead, we evaluate the value of such techniques for reconstructing meaningful DNA sequences in Section VII-F.

D. Noise Effect on Accuracy

Although the pre-processing stage used in the attack reduces the effect of environmental noise and normalizes the acquired signals, significant ambient noise might obfuscate the information leaked in the side-channels, reducing the effectiveness of the attack. Hence, it is important to evaluate the robustness of the trained models against potential environmental noises. To this end, we generate six types of different noises and add them to the recorded raw signals for the test samples: brown noise, which has very high intensity in lower frequencies ($\beta = 2$); pink noise, which has high intensity in lower frequencies ($\beta = 1$); white noise, which has same intensity in all frequencies ($\beta = 0$); blue noise, which has high intensity in higher frequencies ($\beta = -1$); violet noise, which has very high intensity in higher frequencies ($\beta = -2$); and conversation noise, which is a recorded conversation between two persons (the power spectral densities of the color noises are proportional to $\frac{1}{f^\beta}$).

As shown in Figure 10, adding any noise with a high-decibel sound pressure level (SPL) reduces the attack model accuracy. We observe that the noises, which have a medium to high emphasis on their higher frequency components, can mask the leaked signal with lower SPL. If a noise generator is added as a countermeasure against acoustic side channel attacks, then to be effective, it must generate high-frequency noises in close proximity to the synthesizer. General noises in the laboratory, such as employee conversations or air conditioner emission (pink), are unlikely to be effective countermeasures.

E. Microphone Distance Effect on Accuracy

The SPL induced by the DNA synthesizer is inversely proportional to the distance between the DNA synthesizer and the microphone. If an SPL is measured to be L_1 at distance r_1,

it will reduce to L_2 at distance r_2 according to the following equation [25]:

$$L_2 = L_1 - |20log(\frac{r_1}{r_2})|. \quad (6)$$

We used this equation to evaluate the effect of microphone distance to the target DNA synthesizer. During the delivery stages, our experiments yielded an average SPL of 81.15 dB and 77.1 dB for the contact and condenser microphones. We assume that the acoustic signal collected by the contact microphone is free of environment noise. In this scenario, decreasing the SPL of the recorded signal by the contact microphone while adding a constant room noise is equivalent to placing the recorder at a further distance.

Figure 11 shows the result of decreasing the SPL of a recorded signal near the contact microphone in 1 dB increments while keeping the added room noise level constant. Since we know the SPL of the signal within 10 cm of the machine, we add a secondary horizontal axis to the top of the chart to show the distance of the microphone from the DNA synthesizer using Equation 6. If we assume that there is no noise in the environment, the degradation of the SPL of a signal at a further distance would have no effect on the accuracy of a given voting classifier, since a normalization step can revert the SPL back to its expected value; in practice, environment noise is unavoidable. As shown in the figure, increasing the distance decreases the accuracy for all types of classifiers. Once the distance between the microphone and DNA synthesizer exceeds 0.7 meters, the accuracy of all classifiers noticeably degrades, although the exact amount of degradation varies among the classifiers. For example, the Neural Network classifier has a higher accuracy than the Random Forest classifier in close proximity to the DNA synthesizer; however, the Random Forest classifier has higher accuracy when the microphone is placed further away from the synthesizer. The ensemble voting classifier used in the attack methodology can often reach and exceed the accuracy of these two classifiers individually, regardless of the distance.

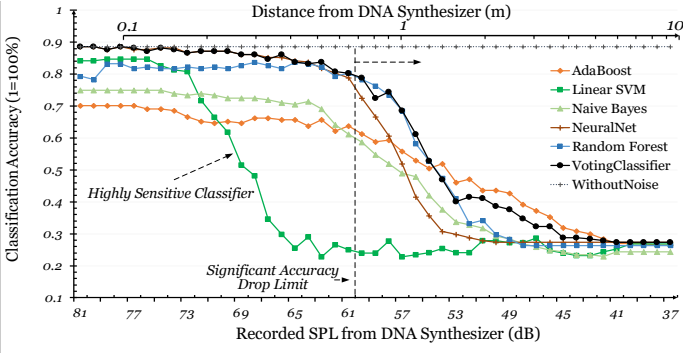


Fig. 11: Effect of microphone distance on various classifiers used in the attack methodology.

F. Test Case Evaluation

This section evaluates the impact of our proposed attack methodology by reconstructing four DNA sequences which were synthesized using the DNA synthesizer shown in Figure 3. To ensure fairness, the test cases considered here were chosen by an author different than the attacker, and the original sequence was provided to the attacker only after the results were submitted for comparison. The attacker makes only one assumption: the sequence is later going to be implanted in a living organism to create some type of protein. Every three oligonucleotide bases translate to a certain amino acid (the building block of a protein) based on this table [57]. As discussed in Section V this assumption may help during post-processing. The DNA sequence test cases evaluated are:

1- Conotoxins. We synthesized part of a DNA sequence that translates to a lethal protein: conotoxin. Conotoxins are recognized by the US Government as potential agents of bioterrorism [26]. We assume that the attacker is potentially a government agency or a similar entity.
2- Human insulin. We synthesized the DNA that encodes the alpha chain of human insulin. Insulin was originally extracted from pig pancreases; in 1979 Goeddel *et al.* added DNA encoding human insulin to bacteria to produce actual human insulin [30]. This was the first major drug produced by synthetic biology and led to the founding of Genentech, a multi-billion-dollar pharmaceutical company.
3 & 4- Peptide. We synthesized two DNA sequences which encoded peptides that were isolated by *in vitro* selection to bind the protein target streptavidin. The peptides have been characterized in [39] and function as high-affinity ligands to the protein target. These peptides could be used as protein affinity tags to purify other proteins from crude cellular lysate or cell-free translation systems.

For the test cases, we use the majority voting rule-based ensemble of classifiers trained in Section VII. We first collect the acoustic signal generated by the machine while synthesizing the aforementioned sequences. After preprocessing and background noise elimination, we manually ensure that the correct signal delivery segments are extracted from the given signals. After segment extraction, the model extracts the same set of features that were used for training. Next, the trained classifier predicts the probabilities for each base type for each delivery. As shown in Table III, choosing the base type with the highest probability results in average accuracies close to the classifier accuracy which is calculated in Section VII. Since we assume that the reconstructed sequences will be used in a biological application, we also provide the number of errors in terms of mispredicted amino acids. We use the K-best sequence algorithm described in Section VI to show the number of trials that an attacker would need to reconstruct the original sequence with perfect accuracy. The results show that, for short sequences, it is possible to reconstruct the sequence with a reasonable number of trials. However, as sequences grow in length, due to the limited accuracy of the classifiers, finding the exact location of the errors in the sequence becomes more difficult. Hence, we conclude that if the number of bases is long enough, achieving 100% accuracy for reconstructing the whole sequence with the given attack methodology solely based on acoustic side-channel data may not be possible. However, failing to reconstruct the sequence with perfect accuracy does not translate to the absolute confidentiality of the data passed to the machine. An attacker can acquire enough information to determine the intended purpose of a reconstructed DNA sequence even if some base predictions are incorrect. As it turns out, reading the sequence of bases in a DNA molecule have always been error-prone.

TABLE III: Results for reconstructing the test cases.

Case #	**Original** Oligonucleotide sequence **Predicted** Oligonucleotide sequence	Sequence Length	Accuracy (%)	BLAST match	Number of guesses to have **N or less** mispredicted amino acid				Brute Force Complexity
					N=3	N=2	N=1	N=0	
1	CGCAA**G**TACTCCTG**C** CGCAA**T**TACTCCTG**A**	15	86.67	Yes	**1**	**1**	**3**	**21**	15x4^15
2	GGAATAGTAGAAG**AA**TGCTGCACAA**G**CATATGCAGCCTA**T**ACGAACTAGAAGAC**T**ACTGCGAC GGAATAGTAGAAG**CG**TGCTGCACAA**T**CATATGCAGCCTA**C**ACGAACTAGAAGAC**G**ACTGCGAG	63	**90.48**	Yes	12	29	>100	>100	63x4^63
3	TGGCGACAT**G**ATAACCCGTCGGA**G**GATCCGGG**G**CG**G**G**GG**CACCTC TGGCGACAT**T**ATAACCCGTCGGA**T**GATCCGGG**T**CG**T**G**TT**CACCTC	45	86.67	Yes	**1**	3	19	>100	45x4^45
4	TTTT**T**CGACCGGT**A**T**GA**T**T**CCGCCCGTGACCCAGGACGCTTGCTT TTTT**G**CGACCGGT**C**T**TC**T**G**CCGCCCGTGACCCAGGACGCTTGCTT	45	88.89	Yes	**1**	3	35	>100	45x4^45

The bold colors with larger font size in oligonucleotide sequence represents the misclassified nucleotide bases

Publicly available software such as BLAST [2] store DNA sequence, their functionality, and can readily determine the most similar known DNA sequences, along with their application, for a given amino acid sequence. If the attacker is a government agency, tools like BLAST may be used to determine if a hostile user is synthesizing DNA sequences that have structural similarities to known hazardous sequences.

In an industrial setting an attacker may work for a competitor whose objective is industrial espionage; in this case, the attacker can use tools like BLAST to derive the likely amino acid sequence of a protein being developed by the company under attack. To quantify the relevant information in the reconstructed sequences, we import amino acid sequences that correspond to the original and reconstructed test cases to the BLAST and then compare output reported for each sequence. Table III summarizes the result of this experiment. If BLAST reports a similar set of candidates for the original and the reconstructed sequences, Table III reports YES in the "BLAST match" column.

VIII. Discussion

A. Attack Cost and its Implications

We spent 56 hours collecting training data on the AB 3400 DNA synthesizer; during this time, human supervision was required for less than one hour in total. We dedicated 5 hours to understanding the structure of the AB 3400 and the scripts used for oligonucleotide synthesis. Manual segmentation of the signal took 20 minutes per synthesis run (less than 3 hours in total); an Intel Core i7-7820X CPU with 16 gigabytes of RAM extracted the features, selected the best features, and trained 6 models in 14 minutes and 28 seconds. Although access to the DNA synthesizer was granted by its operator (a university laboratory) at no cost, the synthesis runs consumed $300 worth of raw chemical materials. Once the models are trained, the attack phase requires 20 minutes to manually segment 60 base deliveries and less than one second per delivery to predict the base.

These costs are negligible in comparison to the cost of real-world drug production in industry. In 2004, for example, the Bill and Melinda Gates Foundation dedicated $42.6 million to the development of an anti-malaria drug [8]. The recipients of the award published their initial results two years later, and completed the research project by the end of the third year. This particular drug was not patented and its recipe was made freely available as a humanitarian gesture; however, the key point is that the drug cost tens millions of dollars to produce, while an attacker could steal its DNA sequence (if kept a secret) in less than one week of time and at a cost of several hundred dollars. This could easily doom a for-profit private sector drug development project.

B. Limitations of Attack Methodology and Experiments

The AB 3400 DNA synthesizer is a widely used commercial product, but is not the only one on the market. The feasibility analysis in Section IV implies that the proposed attack methodology is could be applied to any DNA synthesizer that employs solenoid valves and pipes for chemical delivery; future work will validate this attack methodology against similar machines.

This paper used the same machine for training and validation of the attack model and evaluated its robustness to minute differences between the profiling system and target through the additional of artificial noise (Section VII-D). Further investigation is required to evaluate the effectiveness of the attack model when different AB 3400 machines are used for profiling and targets of the attack.

IX. Countermeasures

Secured structure: One way to prevent acoustic side-channel attacks is to ensure that all the physical components responsible for the delivery stage are similar: identical solenoids, pressure valves, and pipe lengths must be chosen, and they must be placed in a geometrically identical manner; for example, bends in fluid pipes must be identical to eliminate variations in accoustic emissions. Additionally, anti vibration pads (a.k.a. vibration isolators) could be integrated into the inner layers of the DNA synthesizer to reduce the intensity of any emitted observable acoustic noise.

Artificial noise: Redundant physical components may be added to introduce additional noise and decrease the signal to noise ratio, making it difficult to infer the cyber and physical states of the DNA synthesizer. Although intuitive, adding loud noise may bother employees who work in the same environment as the DNA synthesizer. Thus, proper methods, such as those described in Section VII-D, will be needed to search for low intensity noises that can mask information emitted from the acoustic side channel.

Delivery segment obfuscation: The DNA synthesizer's delivery stage is the critical point of vulnerability for this particular

attack. A number of opportunities exist to potentially obfuscate this stage exclusively, such as adding redundant steps of varying time length, which are benign to the DNA synthesis process, and randomly selecting them prior to delivery; however, this may increase DNA synthesis time. Another possibility is to randomly select and execute steps unrelated to base delivery, such as cleaning other pipes or waste disposal, concurrently with base delivery. Although the system will remain observable, the cyber and physical-domain states will be obfuscated, which will increase the difficulty of using data-driven approaches to infer the actual states.

Secured laboratory environment: The most effective practice to assure confidentiality of the synthesized DNA sequences is to prevent any visitor or unauthorized personnel from entering any room that contains a DNA synthesizer. Along similar lines, any unauthorized device found in the same room as a DNA synthesizer should be reported as a security threat. Furthermore, the cyber-security of every electronic device with recording capabilities that enters the laboratory environment, even if authorized, should be considered; the device itself may be compromised by a malicious adversary who can remotely activate acoustic signal recording.

X. Conclusion

We proposed and implemented a novel acoustic side-channel attack methodology on DNA synthesizers to steal the type and order of the bases which were synthesized. We tested our attack model against one of the most widely used DNA synthesizers and showed that ignoring such a confidentiality vulnerability can result in a significant research investment loss. We were able to predict the type of each base delivery with an average accuracy of 88.07%. We introduced a novel way to evaluate the effect of microphone distance without exhaustive experiments which revealed that high accuracy can be expected from the proposed attack methodology with up to a 0.7 m gap between the microphone and the DNA synthesizer. In addition, we showed how a reconstructed sequence can be leveraged using a post-processing approach in a biological domain (such as using the publicly available tool BLAST) to identify the protein that the original sequence intended to encode, despite any errors.

Acknowledgment

This research was supported in part by NSF awards CMMI-1739503 and CMMI-1763795. The authors would like to thank Jen-Yu Liao (University of California, Irvine) for technical assistance with the AB 3400 DNA synthesizer.

References

[1] *Screening framework guidance for providers of synthetic double-stranded DNA*. US Department of Health and Human Services, 2010.

[2] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped blast and psi-blast: a new generation of protein database search programs," *Nucleic acids research*, vol. 25, no. 17, pp. 3389–3402, 1997.

[3] Applied Biosystems, "Applied biosystems 3400 DNA synthesizer: User guide," 2010. [Online]. Available: http://www3.appliedbiosystems.com/cms/groups/mcb_support/documents/generaldocuments/cms_095581.pdf

[4] A. J. Aviv, B. Sapp, M. Blaze, and J. M. Smith, "Practicality of accelerometer side channels on smartphones," in *Proceedings of the 28th Annual Computer Security Applications Conference*. ACM, 2012, pp. 41–50.

[5] R. G. Baraniuk, "Compressive sensing [lecture notes]," *IEEE signal processing magazine*, vol. 24, no. 4, pp. 118–121, 2007.

[6] C. Bayens, G. L. Le T, R. Beyah, M. Javanmard, and S. Zonouz, "See no evil, hear no evil, feel no evil, print no evil? malicious fill patterns detection in additive manufacturing," in *26th USENIX Security Symposium USENIX Security 17)*. USENIX Association, 2017, pp. 1181–1198.

[7] Y. Benjamini and D. Yekutieli, "The control of the false discovery rate in multiple testing under dependency," *Annals of statistics*, pp. 1165–1188, 2001.

[8] Bill and Melinda Gates foundation. (2004) Collaboration of biotech, academia, and nonprofit pharma could significantly reduce cost, boost supplies of antimalarial drug. [Online]. Available: https://www.gatesfoundation.org/Media-Center/Press-Releases/2004/12/OneWorld-Health-Receives-Grant

[9] C. Bolton, S. Rampazzi, C. Li, A. Kwong, W. Xu, and K. Fu, "Blue note: How intentional acoustic interference damages availability and integrity in hard disk drives and operating systems," in *Blue Note: How Intentional Acoustic Interference Damages Availability and Integrity in Hard Disk Drives and Operating Systems*. IEEE, 2018, p. 0.

[10] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[11] H. Bügl, J. P. Danner, R. J. Molinari, J. T. Mulligan, H.-O. Park, B. Reichert, D. A. Roth, R. Wagner, B. Budowle, R. M. Scripp *et al.*, "Dna synthesis and biological security," *Nature biotechnology*, vol. 25, no. 6, p. 627, 2007.

[12] P. Calain, M. C. Monroe, and S. T. Nichol, "Ebola virus defective interfering particles and persistent infection," *Virology*, vol. 262, no. 1, pp. 114–128, 1999.

[13] P. Chapman and D. Evans, "Automated black-box detection of side-channel vulnerabilities in web applications," in *Proceedings of the 18th ACM conference on Computer and communications security*. ACM, 2011, pp. 263–274.

[14] S. R. Chhetri, S. Faezi, and M. A. Al Faruque, "Information leakage-aware computer-aided cyber-physical manufacturing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2333–2344, 2018.

[15] S. R. Chhetri, S. Faezi, and M. A. A. Faruque, "Fix the leak!: an information leakage aware secured cyber-physical manufacturing system," in *Proceedings of the Conference on Design, Automation & Test in Europe*. European Design and Automation Association, 2017, pp. 1412–1417.

[16] M. Christ, A. W. Kempa-Liehr, and M. Feindt, "Distributed and parallel time series feature extraction for industrial big data applications," *arXiv preprint arXiv:1610.07717*, 2016.

[17] S. S. Clark, B. Ransford, A. Rahmati, S. Guineau, J. Sorber, W. Xu, K. Fu, A. Rahmati, M. Salajegheh, D. Holcomb *et al.*, "Wattsupdoc: Power side channels to nonintrusively discover untargeted malware on embedded medical devices." in *HealthTech*, 2013.

[18] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[19] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[20] P. Du, W. A. Kibbe, and S. M. Lin, "Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching," *Bioinformatics*, vol. 22, no. 17, pp. 2059–2065, 2006.

[21] S. R. Eddy, "Hidden markov models," *Current opinion in structural biology*, vol. 6, no. 3, pp. 361–365, 1996.

[22] D. Eppestein, "k-best enumeration," *CoRR*, vol. abs/1412.5075, 2014. [Online]. Available: http://arxiv.org/abs/1412.5075

[23] D. Eppstein, "Finding the k shortest paths," *SIAM Journal on computing*, vol. 28, no. 2, pp. 652–673, 1998.

[24] S. Faezi, "Data-driven modeling for minimizing the side-channel information leakage in additive manufacturing," Master's thesis, UC Irvine, 2017.

[25] F. J. Fahy, *Foundations of engineering acoustics*. Elsevier, 2000.

[26] Federal select agent program. (2017) Select agents and toxin list. [Online]. Available: https://www.selectagents.gov/SelectAgentsandToxinsList.html

[27] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[28] S. Galanie, K. Thodey, I. J. Trenchard, M. Filsinger Interrante, and C. D. Smolke, "Complete biosynthesis of opioids in yeast," *Science*, vol. 349, no. 6252, pp. 1095–100, Sep 2015.

[29] D. G. Gibson, J. I. Glass, C. Lartigue, V. N. Noskov, R.-Y. Chuang, M. A. Algire, G. A. Benders, M. G. Montague, L. Ma, M. M. Moodie, C. Merryman, S. Vashee, R. Krishnakumar, N. Assad-Garcia, C. Andrews-Pfannkoch, E. A. Denisova, L. Young, Z.-Q. Qi, T. H. Segall-Shapiro, C. H. Calvey, P. P. Parmar, C. A. Hutchison, 3rd, H. O. Smith, and J. C. Venter, "Creation of a bacterial cell controlled by a chemically synthesized genome," *Science*, vol. 329, no. 5987, pp. 52–6, Jul 2010.

[30] D. V. Goeddel, D. G. Kleid, F. Bolivar, H. L. Heyneker, D. G. Yansura, R. Crea, T. Hirose, A. Kraszewski, K. Itakura, and A. D. Riggs, "Expression in escherichia coli of chemically synthesized genes for human insulin," *Proceedings of the National Academy of Sciences*, vol. 76, no. 1, pp. 106–110, 1979.

[31] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using networkx," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2008.

[32] M. Hähnel, W. Cui, and M. Peinado, "High-resolution side channels for untrusted operating systems," in *2017 USENIX Annual Technical Conference (USENIX ATC 17)*, 2017, pp. 299–312.

[33] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class adaboost," *Statistics and its Interface*, vol. 2, no. 3, pp. 349–360, 2009.

[34] G. E. Hinton, "Connectionist learning procedures," in *Machine Learning, Volume III*. Elsevier, 1990, pp. 555–610.

[35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[36] A. Hojjati, A. Adhikari, K. Struckmann, E. Chou, T. N. Tho Nguyen, K. Madan, M. S. Winslett, C. A. Gunter, and W. P. King, "Leave your phone at the door: Side channels that reveal factory floor secrets," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 883–894.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[38] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, no. 2. Montreal, Canada, 1995, pp. 1137–1145.

[39] A. C. Larsen, A. Gillig, P. Shah, S. P. Sau, K. E. Fenton, and J. C. Chaput, "General approach for characterizing in vitro selected peptides with protein binding affinity," *Analytical chemistry*, vol. 86, no. 15, pp. 7219–7223, 2014.

[40] S. Li, B. W. Karney, and G. Liu, "Fsi research in pipeline systems–a review of the literature," *Journal of Fluids and Structures*, vol. 57, pp. 277–297, 2015.

[41] M. Loog, R. P. W. Duin, and R. Haeb-Umbach, "Multiclass linear dimension reduction by weighted pairwise fisher criteria," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 7, pp. 762–766, 2001.

[42] I. Martinovic, D. Davies, M. Frank, D. Perito, T. Ros, and D. Song, "On the feasibility of side-channel attacks with brain-computer interfaces." in *USENIX security symposium*, 2012, pp. 143–158.

[43] Mathworks Co. (2017) Matlab r2017b.

[44] L. McBride and M. Caruthers, "An investigation of several deoxynucleoside phosphoramidites useful for synthesizing deoxyoligonucleotides," *Tetrahedron Letters*, vol. 24, no. 3, pp. 245 – 248, 1983. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0040403900813763

[45] A. L. McGuire, R. Fisher, P. Cusenza, K. Hudson, M. A. Rothstein, D. McGraw, S. Matteson, J. Glaser, and D. E. Henley, "Confidentiality, privacy, and security of genetic and genomic test information in electronic health records: points to consider," *Genetics in Medicine*, vol. 10, no. 7, p. 495, 2008.

[46] P. Ney, K. Koscher, L. Organick, L. Ceze, and T. Kohno, "Computer security, privacy, and dna sequencing: Compromising computers with synthesized dna, privacy leaks, and more," in *26th USENIX Security Symposium.[October 25, 2017]*, 2017.

[47] A. Nouri and C. F. Chyba, "Dna synthesis security," in *Gene Synthesis*. Springer, 2012, pp. 285–296.

[48] J. Peccoud, J. E. Gallegos, R. Murch, W. G. Buchholz, and S. Raman, "Cyberbiosecurity: From naive trust to risk awareness," *Trends in biotechnology*, vol. 36, no. 1, pp. 4–7, 2018.

[49] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[50] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and systems magazine*, vol. 6, no. 3, pp. 21–45, 2006.

[51] D.-K. Ro, E. M. Paradise, M. Ouellet, K. J. Fisher, K. L. Newman, J. M. Ndungu, K. A. Ho, R. A. Eachus, T. S. Ham, J. Kirby, M. C. Y. Chang, S. T. Withers, Y. Shiba, R. Sarpong, and J. D. Keasling, "Production of the antimalarial drug precursor artemisinic acid in engineered yeast," *Nature*, vol. 440, no. 7086, pp. 940–3, Apr 2006.

[52] Robert F. Service. (2017) Dna could store all of the world's data in one room. [Online]. Available: http://www.sciencemag.org/news/2017/03/dna-could-store-all-worlds-data-one-room

[53] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[54] M. Schmidt and G. Giersch, "Dna synthesis and security," *DNA microarrays, synthesis and synthetic DNA*, pp. 285–300, 2011.

[55] sckit-learn developers. (2018) Multi-layer perceptron classifier.

[56] Sensaphone. (2017) Sensaphone 1400 environmental monitoring system. [Online]. Available: https://www.sensaphone.com/pdf/LIT-0110_1400Manual_v2.0-1.pdf

[57] J.-J. Shu, "A new integrated symmetrical table for genetic codes," *Biosystems*, vol. 151, pp. 21–26, 2017.

[58] R. Singh. (2014) Synthetic biology market by products (dna synthesis, oligonucleotide synthesis, synthetic dna, synthetic genes, synthetic cells, xna) and technology (genome engineering, microfluidics technologies, dna synthesis and sequencing technologies) - global opportunity analysis and industry forecast, 2013 - 2020.

[59] L. Smith and C. Higgins. (2017) Scholars or spies? [Online]. Available: https://www.insidehighered.com/views/2018/06/26/universities-must-take-steps-protect-american-rd-foreign-agents-opinion

[60] A. Søe-Knudsen and S. Sorokin, "Modelling of linear wave propagation in spatial fluid filled pipe systems consisting of elastic curved and straight elements," *Journal of Sound and Vibration*, vol. 329, no. 24, pp. 5116–5146, 2010.

[61] V. Subramanian, A. S. Uluagac, H. Cam, and R. A. Beyah, "Examining the characteristics and implications of sensor side channels." in *ICC*, 2013, pp. 2205–2210.

[62] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[63] C. Thèves, P. Biagini, and E. Crubézy, "The rediscovery of smallpox," *Clinical Microbiology and Infection*, vol. 20, no. 3, pp. 210–218, 2014.

[64] A. Tijsseling, "Fluid-structure interaction in liquid-filled pipe systems: a review," *Journal of Fluids and Structures*, vol. 10, no. 2, pp. 109–146, 1996.

[65] R. W. Young, "Sabine reverberation equation and sound power calculations," *The Journal of the Acoustical Society of America*, vol. 31, no. 7, pp. 912–921, 1959.

[66] H. Zhang, "The optimality of naive bayes," *AA*, vol. 1, no. 2, p. 3, 2004.

[67] Y. Zhang, A. Juels, A. Oprea, and M. K. Reiter, "Homealone: Co-residency detection in the cloud via side-channel analysis," in *2011 IEEE symposium on security and privacy*. IEEE, 2011, pp. 313–328.

[68] Y. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Cross-tenant side-channel attacks in paas clouds," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 990–1003.